Quartz Services

Session 211

Quartz Services

Mike Paquette
Programmer Archaeologist
mpaque@apple.com

What Can Quartz Services Do for You?

- Control the non-drawing portion of the user experience
- Take over displays for immersive applications
- Reconfigure displays
- Special purpose tasks
  - Reading the frame buffer contents
  - Remote control of a computer
- Most apps don’t need Quartz Services

Overview

- Where does this technology fit in Quartz?
  - Display Management
  - Screen scraping
  - Event injection

Quartz Services

- Display Services
  - Full screen apps support
  - Display configuration control
- Remote Operation Services
  - Screen scraping
  - Posting events for remote control

Quartz Services Definitions

- Display ID
  - A unique identifier for a display
  - Refers to a specific framebuffer and monitor
- Display States
  - Active, asleep, mirrored, on-line, off-line
- CG
  - A unique prefix used on Quartz functions

Display Services

- Display Enumeration
- Display Configuration
- Paletted Display Support
- Display Fade

Competive Technologies

- Emulated Mac OS 9 functions
  - Sprockets
  - QuickTime
  - AGL
  - Display Manager

- Provided to assist in porting apps, and to support CFM apps running in both Mac OS 9 and X

Competive Technologies

- Use CGDirectDisplay instead!
  - Older functions use the Quartz Services functions
  - More precise control when directly using Quartz Services functions

Clients of Display Services

- Games
- Fullscreen applications
- Configuration Tools

New Display Functionality

- Definitions
  - Mirroring
  - HW vs. SW
  - How and when this happens
  - Active vs. on-line displays
- Mac OS X hot-plug display behavior
  - On white iBook, PowerBooks with DVI connector

Fullscreen Access

- Display capture vs. hide menu
  - Bypasses window system
  - More control, but less high level functionality
    - No accessibility support
  - Used by games

Fullscreen Access

- HideMenu
  - Removes menu bar and Dock
  - Your app still runs through window system
  - Used by some presentation apps

Display Capture

- CGCaptureDisplay(CGDirectDisplayID display)
  - Capture one display
  - Display is removed from the window system
  - Forces calling app to foreground
  - Screen may be directly drawn from GL, Quartz, or other sets of functions
- CGCaptureAllDisplays(void)
  - Captures all displays
  - May apply last mode used when the app captured displays

Display Capture

- CGDisplayRelease(CGDirectDisplayID display)
  - Releases a captured display to rejoin the window system
- CGReleaseAllDisplays(void)
  - Releases all displays
  - Restores all display modes, gamma tables
  - May be used with CGDisplayCapture() or CGCaptureAllDisplays()

Drawing To a Captured Display

- CGDisplayGetDrawingContext(CGDirectDisplayID)
  - Produce a Quartz drawing context for a display
  - Context is ‘owned’ by framework; Don’t release it!
- OpenGL
  - Choose a pixel format
  - Set up a full screen GL context
  - When done, tear down GL context
  - OpenGL will usually give you best HW acceleration

Display Configuration

- Transactional model
  - Begin configuration
  - Accumulate operations
  - Apply operations
- System constraints may modify your configuration
  - Layout doesn’t permit overlapping displays
  - Invalid display modes are ignored
  - Some displays may not be unmirrored
- Main display is deepest, largest display at (0,0)

Display Configuration Functions

- CGBeginDisplayConfiguration()
  - Returns a new CGDisplayConfigRef in pointer
  - This ref is passed to the configuration functions to accumulate changes
  - Ref is invalid after being passed to CGCompleteDisplayConfiguration() or CGCancelDisplayConfiguration()

Display Configuration Functions

- CGConfigureDisplayOrigin()
  - Set the origin point of a display
  - An origin point of 0,0 makes this the main display
  - Setting the origin of a display mirroring another display removes the display from the mirror set

Display Configuration Functions

- CGConfigureDisplayMode()
  - Takes a display mode dictionary
  - Sets the display resolution, depth, refresh rate
  - When changing the mode of a display in a mirroring set, other displays in the set whose mode is not being set will be adjusted to contain the bounds of the new mode if possible
  - CGDisplayAvailableModes() returns all display modes
  - CGDisplayBestModeForParameters() to pick a mode

Display Configuration Functions

- CGConfigureDisplayMirrorOfDisplay
  - Make display a mirror of another display
  - Remove display from mirror set by specifying NULL display as display to be mirrored
    - Some displays can’t be unmirrored
    - Driver may choose to do hardware mirroring
  - If displays have different resolutions, a matte may be generated in hardware or software
    - Hardware mattes change display row bytes

Display Configuration Functions

- CGCompleteDisplayConfiguration
  - Complete a display reconfiguration
  - Options control how long settings persist
    - kCGConfigureForAppOnly
    - kCGConfigureForSession
    - kCGConfigurePermanently
      - Non-volatile, per-user settings
    - Deferred configuration changes are applied
- CGCancelDisplayConfiguration
- CGRestorePermanentDisplayConfiguration

Display Configuration Functions

- Configuration may be adjusted by system
  - Check after reconfiguring!
- Functions provided to check for:
  - Display state; active, asleep, on-line, mirroring
  - Display configuration
  - Display vendor, model, serial number
  - Display resolution (pixels per millimeter)
  - CGDisplayUsesOpenGLAcceleration
  - CGDisplayIOServicePort

Display Changed Notifications

- Handled in QD, DM and AppKit
- New Functions at Quartz Services level
  - CGDisplayRegisterReconfigurationCallback
  - CGDisplayRemoveReconfigurationCallback
- Callback is invoked for each display before and after reconfiguration on main/event thread
- Flags passed to callback indicate display changes
- Display is invalid after a kCGDisplayRemoveFlag!

Display Hot-Plug

- Sequence of operations during a hot-plug
  - Restoration of saved configuration
  - Notifications to apps

Display Fade

- The Olde Way—Playing with Gamma Tables
  - CGSetDisplayTransferByFormula
    - CPU intensive
    - Loses ColorSync calibration

Display Fade

- The New Way
  - CGDisplayFade
  - Recommended for performance
  - Produces a smooth transition

Display Fade

- Built-in fade effect for display reconfigurations
  - Default fade to blue, or black if displays are captured
  - 0.3 sec fade out
  - 0.5 sec fade in
- Override with CGConfigureDisplayFadeEffect

```
CGConfigureDisplayFadeEffect(CGDisplayConfigRef configRef,
                             CGDisplayFadeInterval fadeOutSeconds,
                             CGDisplayFadeInterval fadeInSeconds,
                             float fadeRed,
                             float fadeGreen,
                             float fadeBlue);
```

Display Fade

- Reserve and release fade hardware
  - CGAcquireDisplayFadeReservation
  - CGReleaseDisplayFadeReservation
- Reservation good for up to 15 seconds
  - Hold reservation for fade out, fade in

Display Fade

- CGDisplayFade
  - Synchronous or async fade operations

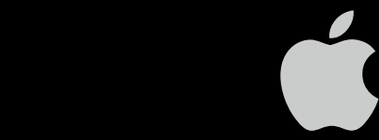
```
CGDisplayFade(myToken,
                    2.0,
 // 2 seconds
                    kCGDisplayBlendNormal,   // Starting state
                    kCGDisplayBlendSolidColor,    // Ending state
                    0.0, 0.0, 0.0,           // black
                    true);
// Wait for completion

CGDisplayFade(myToken,
                   2.0,                      // 2 seconds
                   kCGDisplayBlendSolidColor,     // Starting state
                   kCGDisplayBlendNormal,    // Ending state
                   0.0, 0.0, 0.0,            // black
                   false);
 // Don't wait for completion
```

Paletted Displays

- 8 bit per pixel display modes
  - For legacy application support
  - Minimal or no hardware acceleration available
  - Low precision doesn’t play well with Quartz Compositor
  - Barely supported by current hardware
  - SLOW! 32 bit is fastest on current HW
- Configuration functions allows 8 bit modes to be set
  - Modes cannot be saved as permanent settings
- Palette functions are provided

Paletted Displays

- CGPaletteCreateDefaultColorPalette
- CGPaletteCreateWithDisplay
- CGPaletteCreateWithSamples
- CGPaletteGetColorAtIndex
- CGPaletteSetColorAtIndex
- CGDisplaySetPalette
- CGPaletteRelease



Common Developer Issues

- Recognizing mirroring and display configuration changes
- Testing for mirrored configuration
- Hardware vs. Software mirroring
  - Normally all handled by window system
  - Apps capturing displays must be mirror-aware
- No need to capture displays to reconfigure them!
- Use CGCaptureAllDisplays to avoid desktop layout changes

Demos

- DisplayChanger

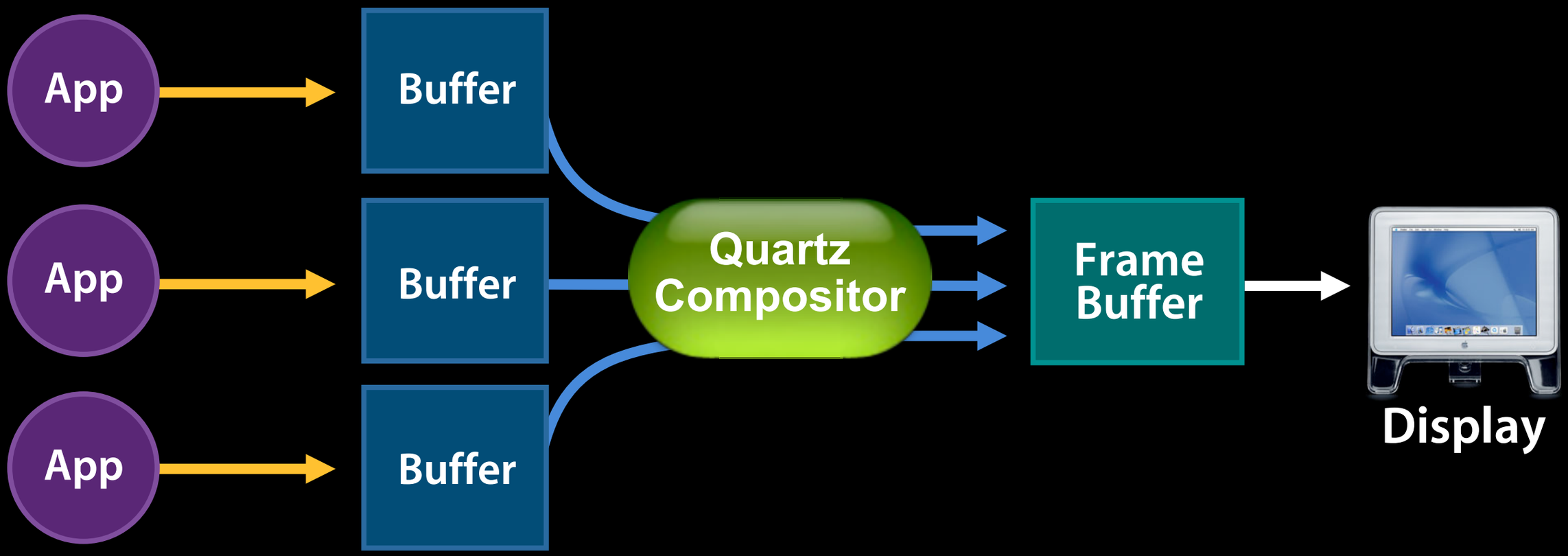
Remote Operations

Remote Operation Services

- What is it?
  - Display drawing activity monitor
  - Generation and posting of events

Architecture of Window Server

- Brief description of backing stores and compositor
- Remote operation services is the best way of getting finished bits of a display update



Architecture of Window Server

- A significant amount of processing may take place between the window's content creation and the bits reaching the display

- Using remote operation services is the best way of getting finished bits of a display update

Clients of Remote Services

- Remote control applications
  - Apple Remote Desktop, Timbuktu, VNC
- Accessibility applications
  - Posting events from specialized devices or apps

Screen Scraping

- “All I get is bits!”
  - Apps draw window content in their own address space, with system or app-specific drawing code
  - Compositor mixes bits from many windows
  - What’s in the app doesn’t match what’s on screen

Screen Scraping Performance

- Async notifications
  - Synchronous behavior would let screen scraper hang the UI
  - Quartz provides notifications or blocking functions to get changes asynchronously
  - Delays in processing just result in changes being coalesced
- New functions in Panther for move notification

Screen Scraping Performance

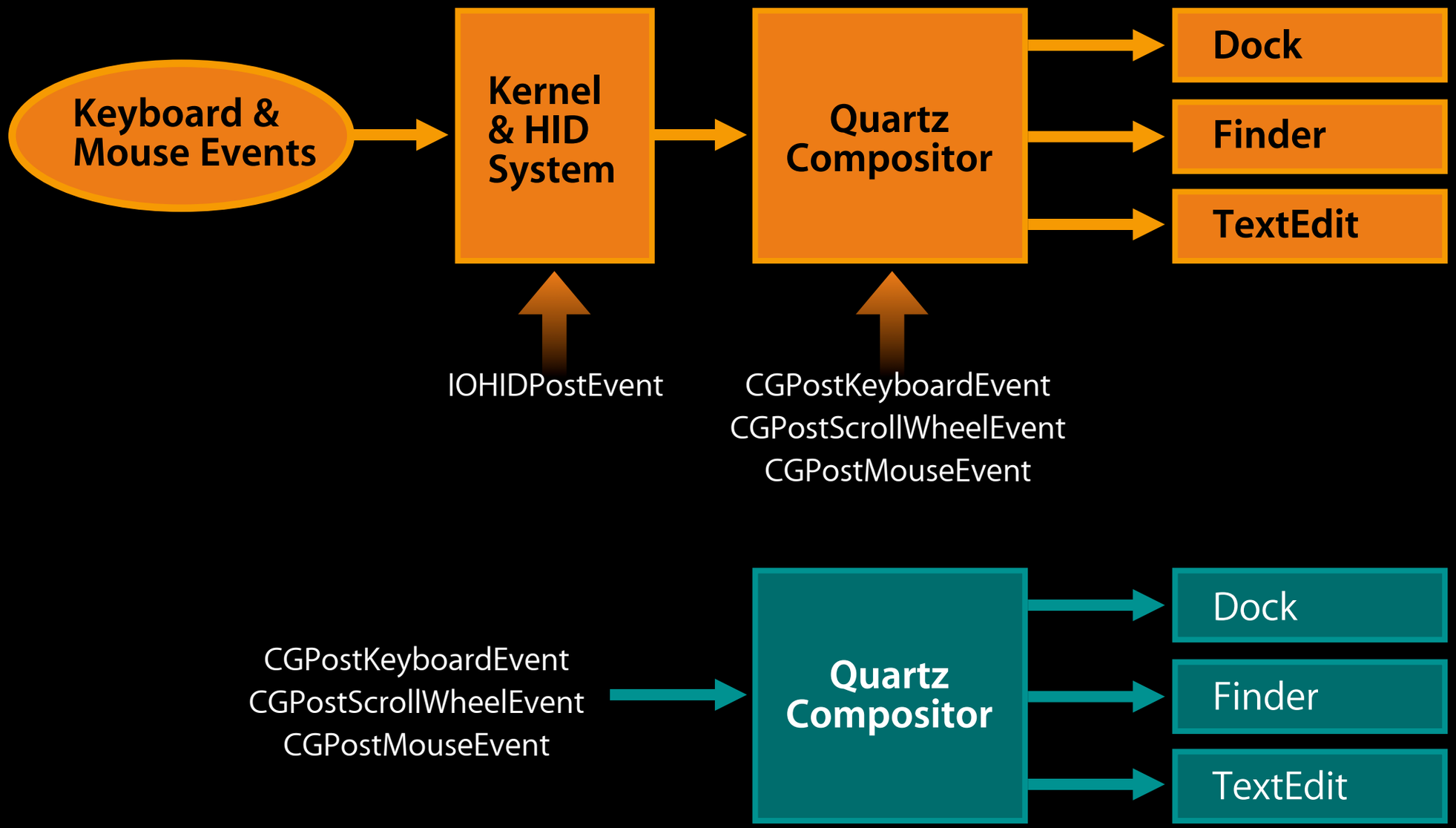
- Use OpenGL to read the screen FAST!
  - Build a full-screen GL context
  - Select screen with glReadBuffer(GL_FRONT)
  - Read screen with glReadPixels()
    - Quadrant 1/Inverted format, so you may need to compensate
    - Format matching hardware will do DMA transfer
- Source code for glGrab demo will be on-line

Demo

- glGrab
  - example code

Event Architecture

- Events flow from IOKit through Quartz to apps



Posting Mouse Events

```
CGError CGPostMouseEvent( CGPoint mouseCursorPosition,
                          boolean_t updateMouseCursorPosition,
                          CGButtonCount buttonCount,
                          boolean_t mouseButtonDown, ... );
```

- Function allows setting desired state
- Appropriate sequence of events is generated to get to desired state
- Don’t forget to post button-up events!
- Move cursor, generating a move event
- CGWarpCursor
  - Move cursor without generating event

Posting Keyboard Events

```
CGError CGPostKeyboardEvent( CGCharCode keyChar,
                             CGKeyCode virtualKey,
                             boolean_t keyDown );
```

- Keycodes vs. charcodes
  - Defined…
  - Most environments ignore the keyChar character code
  - Use virtual key code for best results
- Always post key up
- Modifier flags are generated by posting virtual keycodes for modifier keys

Event Suppression

- Disables local hardware for a short while after posting events
  - Avoids ‘mouse wars’ in Help Desk or classroom use
- Default behavior
  - Local hardware off 0.25 sec after posting event
  - Local mouse off while doing a remote drag
  - Flags from local modifiers combined in posted event
- Force-Quit key sequence reconnects local HW

Event Suppression

```
CGError CGInhibitLocalEvents( boolean_t doInhibit);
CGError CGSetLocalEventsSuppressionInterval(CFTimeInterval seconds);
CGError CGEnableEventStateCombining(boolean_t doCombineState);
CGError CGSetLocalEventsFilterDuringSuppressionState(CGEventFilterMask filter,
                                                     CGEventSuppressionState state);
```

- Functions to override default behavior
  - Always set state as you would want it, just in case the default behavior changes in the future
- Event State Combining controls combining of flags from keyboard modifiers and remote events
  - Turn this off to keep local modifiers out of your event

Configuring Event Suppression

- Most frequent question: “How do I turn it off?”

```
/* Don't block local events after posting ours */
CGSetLocalEventsSupressionInterval(0.0);

/* Let our event's modifier key state blend with local hardware events */
CGEnableEventStateCombining(TRUE);

/* Don't let our events block local hardware events */
CGSetLocalEventsFilterDuringSupressionState(kCGEventFilterMaskPermitAllEvents,
                kCGEventSupressionStateSupressionInterval);
CGSetLocalEventsFilterDuringSupressionState(kCGEventFilterMaskPermitAllEvents,
                kCGEventSupressionStateRemoteMouseDrag);
```

Common Developer Problems

- Filtering all events on the system
  - Not done at the Quartz level
  - Better handled by specialized code such as input managers than as generic low level filters
- Only use functions in the public headers!
  - Undocumented interfaces may go away
  - Undocumented parameters change wildly between releases
  - The stuff in public headers is better

For More Information

Reference Library

Documentation

Quartz Services Reference (Preliminary)

ADC Home > Documentation > Graphics & Imaging >

Quartz 2D Reference

ADC Home > Documentation > Graphics & Imaging >

Drawing With Quartz 2D

ADC Home > Documentation > Graphics & Imaging >

Reference Library

Technical Notes

TN2007: “The CGDirectDisplay API”
ADC Home > Technical Notes >

TN2008: “The CGDirectPalette API”
ADC Home > Technical Notes >

Reference Library

Technical Q&As

QA1011: “How do I use QuickDraw with CGDirectDisplay?”

ADC Home > Technical Q&As >

QA1217: “How can I programmatically determine the DPI of the current video mode?”

ADC Home > Technical Q&As >

QA1218: “How do I tell if a particular display is being hardware accelerated by Quartz Extreme?”

ADC Home > Technical Q&As >

Reference Library

Sample Code

CGFade

ADC Home > Sample Code > Graphics 2D >

ColorBars

ADC Home > Sample Code > Graphics 2D >

ModeWhacker

ADC Home > Sample Code > Graphics 2D >

GLGrab

ADC Home > Sample Code > Graphics 2D >

Fullscreen CGL

ADC Home > Sample Code > Graphics 3D >

Reference Library

Other Resources

OpenGL Programming Guide (3rd Edition)
http://www.opengl.org/developers/documentation/books.html

OpenGL Reference Manual (3rd Edition)
http://www.opengl.org/developers/documentation/books.html

Q&A

Travis W. Brown

Graphics & Imaging Evangelist

travis@apple.com

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference